# SELF-CONTAINED APPLICATION DISK FOR AUTOMATICALLY LAUNCHING APPLICATION SOFTWARE OR STARTING DEVICES AND PERIPHERALS

## RELATED APPLICATIONS

[0001] This application is a continuation-in-part (CIP) of application Serial No. 10/367,650 filed on February 14, 2003 which is a continuation of Serial No. 09/529,992 filed July 26, 1999, now U.S. Patent No. 6,529,992.

## TECHNICAL FIELD

[0002] The present invention relates to application software contained on removable storage media. In particular, the present invention relates to a system wherein instructions to configure operating system environmental variables and to load predetermined application software are stored on removable media such that the application and its associated data files are automatically launched for use upon insertion of the removable disk into a removable media drive.

## BACKGROUND

[0003] In the beginning of the personal computing era, software applications were simple and small mainly because of limitations in storage and computing power. For example, one

of the first widely used software applications, VisiCalc, was only 25 kB in size and was typically run from a floppy disk. Because of these limitations, data files created by VisiCalc were often stored on the same floppy disk as the program file. However, as processing power and capabilities of personal computers increased, additional features were added to software applications to take advantage the increased processing power. These features have lead to a tremendous increase in the size of software applications. For example, Microsoft® Excel97 now has an executable file that is approximately 5.4 MB, which is an increase of 21,600% over the early VisiCalc spreadsheet software. In addition, this figure fails to account for the shared files that Microsoft® Excel97 requires for proper execution. Because of the increase in size and complexity of modem software, applications are no longer developed to be executed from a single piece of removable media, and at a minimum, must be partially installed to the computer's hard disk drive for use.

**[0004]** While modem software applications have greater functionality and are easier to use than their early counterparts, they have created usability problems for end users. A first problem is that managing data files has become increasingly difficult. With personal computer hard drives exceeding 20 GB in storage capacity, users can store almost

everything they generate and download. Naming, organizing and differentiating between data files is extremely difficult, particularly in view of the common practice of users to name and save files using cryptic or uninformative names (e.g., today.doc, temp.xls, or junk.txt)..

**[0005]** Another problem has been created by the shear number of personal computers an peripherals devices. A typical user may have a computer at home, a computer at work, and a portable computer (e.g., notebook) that he or she uses every day. Thus, if a user wants to use a particular data file(s) on more than one computer, the user must manually copy the data file(s) to a removable disk and either access the data file(s) from the removable disk or copy them to the other computer. In addition, each computer that the user wants to use must have the particular piece of software that generated the data file(s) installed thereon in order to properly access the data. Inevitably, as users move data from one computer to another, data is lost as newer files are overwritten by older versions. Further complicating file management tasks is when the user creates several pieces of removable media containing several versions of the data files and then must figure out which version is the correct version by looking at the time stamps or inspecting the contents of each of the data files. Peripheral devices and their associated software have become

numerous and are in many cases difficult to use. In addition, many peripheral devices create data files that must be saved for use by other application software.

**[0006]** There have been some advances in file management. These include file managers and applications that synchronize files between computers to reduce the effort necessary to maintain current versions of data files on multiple computers. However, these programs are severely limited because they typically cannot synchronize files between more than two computers. In addition, the files to be synchronized are often manually selected by the users. Thus, if a particular piece of software creates and access multiple data files, the user must manually select all of the data files for synchronization between two computers, otherwise errors may occur. Often users forget to select a particular data file for synchronization and later find that they are unable to utilize any of the data on another computer. This problem not only affects synchronization programs, but also occurs when users manually copy data files to removable media, in the scenario noted above.

**[0007]** There have been other advances in making computers easier use, such as graphical user interfaces and applications that automatically launch programs contained on removable media. An example of an application that

automatically launches another application upon insertion of the removable media into a computer and include the Windows® "Autorun" feature which automatically runs an application contained on, e.g., a CD-ROM. When the CD-ROM is inserted into the drive, the operating system checks for a file named "autorun.inf" in the root directory of the CD. Within the autorun.inf file is a pointer to an executable file named "autorun.exe" which launches the software on the CD. Typically, the Autorun feature is used to launch software setup/install routines to install software contained on the CD to the hard drive. However, the Autorun feature is limited in that it does not configure a computer to run application software based on instructions contained on the removable media, nor does the Autorun feature have the ability to manage events such as ejection of the removable media, closing of the application running from the removable media, and storage of data files to the removable media.

**[0008]** However, even with the above advancements, computers remain difficult for many people to use. For example, when a user turns on the computer, he or she is most often presented with the graphical user interface. At this point, it is up to the user to navigate to and launch the application software that he or she desires to use. Although this task seems relatively easy, it is not for many users as

there is no standardization on where a particular application software is located on the computer. Clearly, the personal computer has failed to achieve the simplicity of other home appliances, which has prevented large numbers of people from accepting and using computers.

**[0009]** In view of the above, there is a need for a method and apparatus for facilitating the maintenance and use of a user's data, application software and computer devices. Further, there is a need for a method and apparatus that performs these functions automatically with little or no input from the user in order to prevent accidental loss of data and to provide ease of application use. Such a need would be solved by an self-contained application stored on removable media that is adapted to automatically start devices or launch application software from the removable media or the computer's hard drive upon insertion into the computer, while also saving the data files to the removable media or other predetermined location. The present invention provides such a solution.

## SUMMARY

**[0010]** In view of the above, the present invention, through one or more of its various aspects and/or embodiments is thus presented to accomplish many advantages, such as those

noted below. One embodiment of the present invention includes a computer-readable medium containing instructions to be executed by a computing device when the computing device is coupled to a media drive that communicates with said computer-readable medium. The instructions comprise configuring the computing device in accordance with an instruction file contained on said computer-readable media and launching a participating application in accordance with information in said instruction file. The computing device monitors events in the media drive and the participating application until execution of the participating software application is detected to be completed or terminated. Once the application is completed, certain data files are saved to a predetermined location specified in said instruction file and temporary files deleted to unconfigure said computing device.

**[0011]** Another embodiment of present invention is a method of managing a plurality of software programs for use with a computer device in accordance with special instructions contained on storage media external to or removable from the computer device. The method comprises activating the storage media to establish communication with the computer device and providing a list of the plurality of software programs located on the storage media. The user may select one of the plurality of software programs for execution, wherein special

instructions are provided in a control file located on the storage media for each of the selected software programs. The special instructions include configuration information, software launching information; and data file storage information which are used to transfer files and configure the computer device from a first state in accordance with the configuration information to a second state after the selected software is launched. The software is monitored to determine various stages in the operation of the selected software, and upon termination of the software program the computing device is unconfigured by removing the files transferred to the computing device to essentially return the computer device to the first state.

## DESCRIPTION OF DRAWINGS

**[0012]** The foregoing summary, as well as the following detailed description of the preferred embodiments, is better understood when read in conjunction with the appended drawings. For the purpose of illustrating the invention, there is shown in the drawings an embodiment that is presently preferred, in which like reference numerals represent similar parts throughout the several views of the drawings, it being understood, however, that the invention is not limited to the

specific methods and instrumentalities disclosed. In the drawings:

**[0013]** FIG. 1 is a block diagram of the components of a personal computer in which the present invention is embodied;

**[0014]** FIG. 2 is a block diagram of the components of a preferred removable media drive shown in FIG. 1;

**[0015]** FIG. 3 is a flow chart illustrating an overview of the processes performed by the present invention; and

**[0016]** FIG. 4 is an exemplary task disk control file in accordance with the present invention.

**[0017]** FIG. 5 is a flow chart illustrating an overview of a process to manage multiple applications with the present invention.

## DETAILED DESCRIPTION

**[0018]** The present invention provides a novel application on a removable disk used within a computing environment to maintain application software, data and devices. In accordance with the present invention, a user inserts the removable disk containing the application software to be executed into the personal computer, and the software application or device is automatically launched/started and readied for use. The present invention advantageously allows a

user to maintain everything he or she needs using a removable disk.

**[0019]** Referring now to FIGS. 1-2, exemplary hardware in which the present invention may be embodied will be described. As illustrated, the preferred platform is a personal computer (PC) 20, which may comprise Windows® 95/98 or Windows®. Workstation-based personal computer having, e.g., an Intel Pentium( processor or higher, a long-term non-removable storage device (e.g., a IDE or SCSI hard disk), a removable media drive (e.g., CD-R, CD-RW, DVD, or other removable floppy or hard disk drive), random access memory (RAM), communication peripherals (e.g., network interface card, modem, and/or terminal adapter), and suitable application programs (e.g., Dial-up networking software and a Web Browser).

**[0020]** As shown, the PC 20 may be divided between internal and external components. The internal components include a Basic Input/Output System (BIOS) 70 and a processor (CPU) 66 that control the overall functioning of the PC 20. Memory 64, a hard disk drive 76, a floppy disk drive 74, a tape drive 78, a CD-ROM drive 80, a MODEM/Terminal Adaptor/Network Interface Card 82, and a removable media drive 52a are also connected to the CPU 66. The removable media drive 52a or 52b operates to read and/or write to a storage

media contained within a removable storage cartridge 28. The exemplary PC 20 of FIG. 1 is configured with two removable media drives 52a and 52b to emphasize that a removable media drive can be implemented in either internal or external form.

**[0021]** By way of a non-limiting example, the removable media 28 may comprise a ZIP® disk manufactured by Iomega Corporation, Roy, Utah. Each Iomega ZIP® disk contains a unique serial number that is written to a predetermined track during the formatting process which may be used as the unique identifier. The serial number is preferably created by but not limited to a pseudo random number generator. Further, while the media 28 has been described in terms of a ZIP® disk, it is not limited to the ZIP® disk, as the use of other removable media types having a unique serial number is within the scope and spirit of the present invention such as CD-R, DVD-RAM, and other removable floppy and hard disks.

**[0022]** The MODEM/Terminal Adaptor/Network Interface Card 82 may comprise individual cards performing communications-related functions, as known in the art. The MODEM/Terminal Adaptor/Network Interface Cards 82 are included within PC 20 to provide communications to external networks to which the PC 20 is connected. In particular, the MODEM/Terminal Adaptor/Network Interface Card 82 may be used to access LAN 14, ISP 18 and network infrastructure 12.

**[0023]** Communications between internal and external devices may be accomplished via controllers provided within the PC 20. A serial/parallel/USB port controller (which may comprise separate controllers) 58, a monitor controller (video card) 60, and a keyboard and mouse controller 62 each provide an interface between the CPU 66 and an external removable media drive 52b (or printer), monitor 54, and keyboard and mouse device 56, respectively. A hard disk and floppy disk controller 72 serves as an interface between the CPU 66 and the hard disk 76 and the CD-ROM drive 80, and the floppy disk 74 and tape drive 78, respectively. It will be appreciated by those skilled in the art that the disk controller 72 may comprise separate floppy and hard disk controllers (e.g., IDE or SCSI controller).

**[0024]** A removable media controller 68 serves as an interface between the removable media drive 52a and the CPU 66. For example, the removable disk controller 68 may comprise a Small Computer System Interface (SCSI) or Integrated Drive Electronics (IDE) interface controller. A hard disk and floppy disk controller 72 serves as an interface between the CPU 66 and the hard disk 76 and the CD-ROM drive 80, and the floppy disk 74 and tape drive 78, respectively. Alternatively, the removable media drive 52a may utilize the disk controller 72 as an interface to the CPU 66.

**[0025]** Referring now to FIG. 2, there is illustrated a block diagram of an exemplary media drive 52 having a SCSI interface to the PC 20 (via controller 68). The media drive 52 preferably comprises, a ZIP® drive, manufactured by Iomega Corporation, Roy, Utah; however, the present invention is not limited to such removable media drives and other media drives may be used as media drive 52. Examples of other media and drives that may be used with the present invention include hard disk drives, flash memory drives, including CompactFlash drives, USB thumb drives, optical drives, network drives, and any other media that may be used to store information such as data and programs. The media drive 52 includes components that provide for communication between the read/write channel for the media (lower right side of diagram) and the PC 20 (upper left side of diagram). The media drive 52 includes an AIC chip 101 which performs the SCSI 102, the direct memory access (DMA) 103, and disk formatter 104 functions. The interface also includes a PHAEDRUS 105 which includes an 8032 microcontroller 106, a 1 kByte RAM 107 and an application specific integrated circuit (ASIC) 108. The ASIC 108 may perform various functions, such as servo sequencing, data splitting, EOC, ENDEC, A-to-D, and D-to-A conversion. The communication between the media drive 52 and the PC 20 is accomplished through transfers of data between the

input/output channel of the media drive 52 and the media controller 68 (e.g., SCSI controller) of the PC 20.

**[0026]** Referring now to FIG. 3, there is a flow chart illustrating the processes performed by the present invention. As will become evident to those of ordinary skill in the art, the features and aspects of the present invention discussed below may be implemented by any suitable combination of hardware, software and/or firmware. Preferably, the present invention runs as a background task or service that recognizes and reads a task disk control file stored on the removable media upon detection of insertion of the removable media into the computer. The task disk control file includes instructions to configure the computer operating environment and launch application software resident on the removable media or the computer's hard disk (or on both). Also, the task disk control file may be used to start application software that controls computer devices and/or the devices themselves. Upon closing the predetermined application software or device, data files are saved in accordance with instructions in the control file, the computer's environment is cleaned-up and the removable media is ejected. As will be described below, the system of the present invention serves to greatly simplify the use and operation of computer application software, devices and peripherals because users are not required to have any

knowledge of how to launch the application or start the device or peripheral they wish to use. In other words, all the user needs to know is that the application, device or peripheral he or she wants to use will be automatically launched/started upon insertion of the removable media.

**[0027]** The details of the present invention will now be described with reference to FIG. 3. An application or system service running on the PC 20 monitors device commands to and from the removable media drive 52 (step 200). It is preferable that this be a background application or service that is loaded when the computer is booted-up. An example of such an application to monitor device commands to the removable media drive 52 is IOwatch, available from Iomega Corporation, Roy, UT. The application or service resides on the computer 20 and automatically, and without requiring any user intervention, recognizes the insertion of a piece of removable media.

**[0028]** At step 202 it is determined if a monitored device command is "a media change condition." Media change conditions indicate that removable media 28 has been inserted in to the removable media drive 52 or that some action has occurred with regard to the removable media 28 currently in the drive 52. If no media change condition is present at step

202, then processing returns to step 200 to monitor for subsequent device commands.

**[0029]** If a media change condition is present at step 202, then the present invention begins its processing by identifying an event type associated with the media change condition. Events that may be identified and routed at step 204 include, but are not limited to: device events and events resulting from calls from a "participating application" 201. A "participating application" 201 is defined herein as a software application that has knowledge of, and cooperates with, the present invention by making and accepting event calls to and from the present invention. The participating application may be located on, e.g., the hard disk 76, the removable media 28, a CD-ROM, or network location. In addition, participating applications may control peripherals and devices attached to or within the computer (e.g., tape drives, scanners, etc.). Media change conditions are provide to the present invention by known mechanisms such as inter-application messages, program calls, and event notifications.

**[0030]** A first identified event may be a media insertion event (step 206), which is a condition that exists when the removable media 28 is inserted into the removable media drive 52. The first identified event may also be a user launching a program in a typical manner, such as by double

clicking on an icon or typing the program filename in a RUN command. When the first identified event is a media insertion event, the media may be detected as inserted by the drive in accordance with the method of U.S. Pat. No. 5,854,719, to Ginosar et. al., which is incorporated herein by reference in its entirety.

**[0031]** A media insertion event may be defined as detecting a drive containing media (whether fixed or removable) being connected to the system. For example, a drive may connect to a computer system through the USB interface. Upon insertion of the USB cable into the system, the drive is detected and activated. The insertion of the USB cable may be defined as the media insertion event. Of course, other interfaces such as Firewire or SCSI may be used in place of a USB connection. By detecting the insertion of a cable such as a USB cable as a media insertion event, the present invention may be used with both fixed and removable media. If a drive using removable media is attached to the system without a removable media cartridge being inserted, a media insertion event will occur, but because no participating application will be found, the process will continue to monitor for an additional media insertion event. If a hard drive or other fixed media drive is attached to the system containing participating applications, then the process with

continue as described below. Thus, by defining a media insertion event as either the insertion of removable media or attachment of a drive (such as via the USB port), the present invention may be used with a variety of media drives and styles. Once the media insertion event is detected, the process continues to step 208 where the system checks for the presence of a task disk control file (TDCF) 220 on the media 28.

**[0032]** An exemplary task disk control file 220 is illustrated in FIG. 4, and contains configuration and execution information and regarding the participating application. For example the TDCF 220 may contain information such as the executable filename, path information, and Windows® registry data, including the identification of necessary device drivers, Dynamic Link Libraries, VxDs, etc. The contents of the TDCF will be discussed in greater detail below.

**[0033]** If a valid task disk control file (TDCF) 220 does not exist (step 210), the disk is mounted for use, however, the system returns to step 200 to monitor for subsequent device commands, as the disk cannot be used to configure and launch an application or device. Alternatively, a message may be generated and displayed to the user indicating the TDCF 220 is not valid. If a valid TDCF 220

exists, then at step 212, a unique identifier (e.g., serial number) of the media 28 is read. In accordance with a feature of the present invention, the TDCF 220 may be encrypted using the media serial number as an encryption key to provide an added measure of security and to prevent the application software from being copied and run from another piece of media. For the purposes of the present invention, any encryption algorithm may be used.

**[0034]** The media serial number may be obtained by an application running on the PC 20 that reads the unique identifier and authentication code. This is performed by querying the media using an application programming interface (API) such as the Iomega Ready API, or other suitable method.

**[0035]** The Iomega Ready API when invoked causes the media drive to read the unique serial number from the predetermined track by using the SCSI 0.times.06 Non-Sense Command. In particular, by invoking the Disk Status Page (page 0.times.02) of the Non-Sense Command, the media serial number may be determined by reading offset bytes 20-59 of the returned data structure. Exemplary source code for reading the serial number of an Iomega ZIP® drive and disk is as follows:

```
void CClientApp::GetZipDrive( )
{
     int j,k;
     m_DriveNum=0;
     for(j=0;j<26;j++)
          //scan the drives and find the IOMEGA drives
```

```
        {
            if(IsIomegaDrive(j))
            {
                k=GetGeneralDevType(j);
                if( k==DRIVE_IS_ZIP)
                {
                    m_DriveNum=j;
                j=26;
                }
            }
        }
    }
    void CClientApp::GetSerialNumber( )
    {
    unsigned char szBuffer[1024];
        memset(szBuffer,0,sizeof(szBuffer));
        memset(&m_SerialNumber,0,40);
        GetInfoNonSense(m_DriveNum,0.times.02,szBuffer);
        memcpy(&m_SerialNumber,&szBuffer[22],39);
    }
```

**[0036]** It can be appreciated that the unique serial number should contain a sufficient number of bits (length) to ensure that no two pieces of media have the same identifier. For example, each Iomega ZIP®. disk contains a unique 39 byte (312 bits) serial number, and other bit lengths may be utilized.

**[0037]** Once the media serial number is read, the task disk control file (TDCF) 220 is processed at step 214, where it is decrypted using the media serial number as a decryption key. It is noted that encryption of the TDCF 220 is not necessary for the operation of the present invention, and is provided as a security feature to prevent, e.g., illegal copying of copyrighted program files on the removable media 28. Accordingly, step 210 may be provided with a mechanism to

determine if the TDCF is encrypted, and if not, jump to step 216.

**[0038]** At step 216, the unencrypted TDCF 220 is read and the system environment is configured in accordance with information contained in the TDCF 220 prior to launching the participating application. The TDCF 220 may contain information for one application as described below with reference to FIG. 4, or may contain information for a plurality of applications. Further details for configuring the system environment from the TCDF 220 are explained below as described with reference to FIG. 5.

**[0039]** Referring to FIG. 4, the TDCF 220 contains three sections of configuration information labeled as follows: pre-run, application run, and clean-up. The pre-run section configures registry information such that the operating system is aware of the path, environmental variables and commands to launch the participating application. Optionally, the pre-run section may specify files (e.g., executable files, DLLs, device drivers, etc.) that are to be copied to the hard drive to properly execute the participating application. The application run section instructs the operating system to load the participating application, which launches the participating application for use by a user (see, step 218, below). The clean-up section removes all registry

information added by the pre-run section (see, step 224, below). Optionally, the clean-up section may remove from the hard drive the files (e.g., executable files, DLLs, device drivers, etc.) copied thereto to execute the participating application.

**[0040]** In accordance with the present invention, the TDCF 220 may configure the system environment to run applications in several modes. In a first mode of operation, the participating application is stored on, and run from, the removable media 28 upon insertion. In this first mode of operation, everything necessary to run the participating application is maintained and executed from the removable media 28. Thus, no files are copied to the hard drive, nor are any files required to be on the hard drive to run the participating application.

**[0041]** In a second mode of operation, the participating application cannot be run from the removable media 28 alone. In this mode, certain files are copied to the hard disk as specified in the pre-run section of the TDCF 220. These files may include, but are not limited to the above-mentioned executable files, DLLs, device drivers, etc. It may be necessary to copy files to the hard disk to because of limitations in speed or size of the removable media 28. In the second mode, when the participating application is closed or

the removable disk 28 ejected, the previously copied files are removed from the hard drive in accordance with the clean-up section such that the hard disk is returned to essentially the same state it was in prior to the insertion of the removable media 28.

**[0042]** In a third mode, the participating application cannot be run from the removable media 28 alone, as in the second mode. However, in this mode when the participating application is closed or the disk ejected, the files are not removed from the hard drive. In the third mode, the previously copied files remain on the hard disk in accordance with the clean-up section, such that any subsequent insertion of the removable media 28 would not require copying of the specified files to the hard drive. This third mode of operation speeds up the launching of the participating application for second and subsequent insertions of the removable disk 28 into the removable media drive 52.

**[0043]** In a fourth mode, the participating application is installed to, and stored on, the computer's hard drive. The participating application's installation program preferably includes an option to make it "participating" in accordance with the present invention. Optionally, a plug-in may be used to hook into the participating application to make it "participating" in accordance with the present invention. The

pre-run section of the TDCF contains all of the instructions necessary to automatically launch the participating application from the hard drive and, optionally, load data files as the participating application is launched. The data files are preferably on the removable media 28, but also may reside on the hard disk. In this fourth mode, the TDCF 220 contains a list of instructions to open and close the participating application, such that the removable media acts as a "key" to automatically open the participating application upon insertion into the removable media drive 52.

**[0044]** It is noted that the TDCF 220 of FIG. 4 is provided herein for exemplary purposes only. Further, the TDCF 220 may contain additional or fewer instructions to configure the computer 20, copy files to the computer 20, etc. than illustrated. Further, additional or few modes of operation are possible. In accordance with the present invention, the TDCF 220 contains the instructions necessary to properly configure and launch the participating application upon insertion of the removable media 28, and gracefully remove configuration settings and save data files without the need for user intervention.

**[0045]** Returning again to FIG. 3, at step 218, once the PC 20 is configured in accordance with the TDCF 220, the

participating application is launched for use and the system returns to step 200 to monitor for new events.

**[0046]** Returning again to step 204, if an event identified is a Save/Close Complete event, then processing continues at step 222. A Save/Close Complete event occurs when the user has finished using the participating application and either selects exit or clicks the "x" box (i.e., close window box) in the application window. At step 222 the participating application saves all data files, deletes all temporary files and indicates to the system that it is ready to terminate. The data files may be saved in accordance with a predetermined file path set by the TDCF 220. At step 224 all configuration information is removed from the PC 20 in accordance with the clean-up section of the TDCF 220. The clean-up section will perform certain clean-up functions based on a mode of operation, as noted above. Next, at step 226, the participating application terminates and the removable media 28 is ejected by the present invention from the media drive 52. Processing then continues at step 200.

**[0047]** Again returning to step 204, if an event detected a Media Eject Request Event, then processing continues at step 228. A Media Eject Request Event is a condition that exists when the user has initiated an eject function by pressing an eject button provided on the drive 52

or through a provide software eject feature. At step 230, the participating application is notified via inter-application mechanisms, such as a message, that a "request for media eject" was detected. Next, at step 232, the system executes steps 222-226 as discussed above to complete. Once the present invention is notified of a completion at step 234, processing returns to step 200 to monitor for new events.

**[0048]** In an alternative embodiment, the TDCF 220 may control a plurality of applications. This embodiment may be used, for example, in a hard disk drive that is attached to a computer. Under this embodiment, the TDCF 220 allows the user to select one or more of the plurality of applications for execution. The TDCF 220 may present the selections to the user in a menu using text, a graphical user interface, or any other system enabling the user to make a selection. Once the user selects to run one of the programs on the media, the TDCF 220 executes the selected program according to the parameters of the TDCF 220 or a sub-TCDF.

**[0049]** FIG. 5 further illustrates the process of configuring the system environment from the TDCF 220 as performed in step 216. The process begins in START block 300. Proceeding to step 305, the process examines the TDCF 220 to determine whether it is a single application TDCF 220 or a multiple application TDCF. If the TDCF is a single

application control file, the process proceeds along the YES branch to step 310. In step 310, the process returns to step 218 of FIG. 3 to launch the application according to the instructions in the TDCF 220. Returning to step 305, if the TDCF is a multiple application control file, the process proceeds along the NO branch to block 315.

**[0050]** In block 315, the control file determines all participating applications present on the media. Each participating application may be defined in the control file, or the control file may scan the media to find appropriate applications. Instructions for executing each of the participating applications may be included in the control file, or each of the applications may have a corresponding sub-TDCF. The sub-TDCF may contain similar information as described in FIG. 4.

**[0051]** Proceeding to block 320, the process presents the list of participating applications to the user for selection. The process may present the list for selection using a variety of techniques without departing from the spirit of the invention. For example, the programs may simply be presented as a text selection list, or a menu system may be used. Also, the process may incorporate features of the operating system for present the applications for selection. For example, in a system running Microsoft Windows, a

graphical user interface using the Windows desktop and program icons may be used.

**[0052]** Proceeding to block 325, the process monitors the system to determine if an application is selected for execution. If no application is selected, the process proceeds along the NO branch in a loop back to block 325 to continue to monitor for a selection. If a selection is made, the process proceeds along the YES branch to block 330.

**[0053]** In block 330, the selected application is launched according to the details of the control file or the sub-TDCF as appropriate. After the application is launched, the process returns to block 325 to monitor for additional applications to be selected. Simultaneously, the process proceeds to block 335 to monitor whether the selected application is saved/closed or if a complete event such as media removal occurs. If the application simply continues to function, the process proceeds along the NO branch to remain in a monitoring loop. However, once a complete event is detected, the process proceeds along the YES branch to block 340.

**[0054]** In block 340, the clean-up section of the TDCF 220 is executed and the program removes all appropriate files from the main system and saves the necessary files on the removable media. As stated above, the clean-up section may

remove all traces of the program operation from the main system. After the clean-up section of the TDCF 220 is completed, the process proceeds to block 345.

**[0055]** In block 345, the process determines if the media is still communicating with the main system, such as the removable media still being inserted into the drive. If the media remains present and running, the process proceeds along the YES branch to continue to monitor for selection of applications. If the media has been removed, the process proceeds along the NO branch to terminate in the END block 350.

**[0056]** Thus, as described above, the present invention provides a new and improved system for managing data files and applications that requires very little, if any, user attention. In accordance with the present invention, the user need only know that insertion of a particular piece of removable media launches a particular application or starts a particular device or peripheral that he or she wants to use. The user's data files are saved to the same piece of removable media or other specified location for easy retrieval, therefore, the user's data is easily obtainable for use when the application is launched. When the user has completed using the application and exits, the removable media is ejected from the computer to be placed away for safekeeping.

**[0057]** It is noted that the foregoing examples have been provided merely for the purpose of explanation and are in no way to be construed as limiting of the present invention. While the invention has been described with reference to preferred embodiments, it is understood that the words which have been used herein are words of description and illustration, rather than words of limitations. Further, although the invention has been described herein with reference to particular means, materials and embodiments, the invention is not intended to be limited to the particulars disclosed herein; rather, the invention extends to all functionally equivalent structures, methods and uses, such as are within the scope of the appended claims. Those skilled in the art, having the benefit of the teachings of this specification, may effect numerous modifications thereto and changes may be made without departing from the scope and spirit of the invention in its aspects.